

УДАЛЁННОЕ УПРАВЛЕНИЕ ПЛАТФОРМОЙ ДЛЯ ВНЕКОРАБЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ НА ОСНОВЕ RASPBERRY PI

А.В.Бышенко студентка гр 1А01

Томской политехнический университет

М.В.Бердник ученик 011 группы

Лицей при ТПУ

Тел: +79234096671

E-mail: fyrufjghjkloool@gmail.com

Ежегодно проводятся международные инженерные соревнования школьных команд «Кванториада», в которых проекты разделяются по трекам. Команда Детского технопарка «Кванториум» города Томска реализует проект в треке «Платформа для внекорабельной деятельности». Платформа представляет собой устройство, способное обеспечивать решение некоторых задач при выходе космонавтов в открытое космическое пространство. Платформа автономная, перемещается по рельсу с помощью колёс в необходимое место. Движение колёс требует внешнего управления. На рисунках 1 и 2 приведены структурная схема платформы и её 3D-модель.

Структурная схема платформы для ВКД

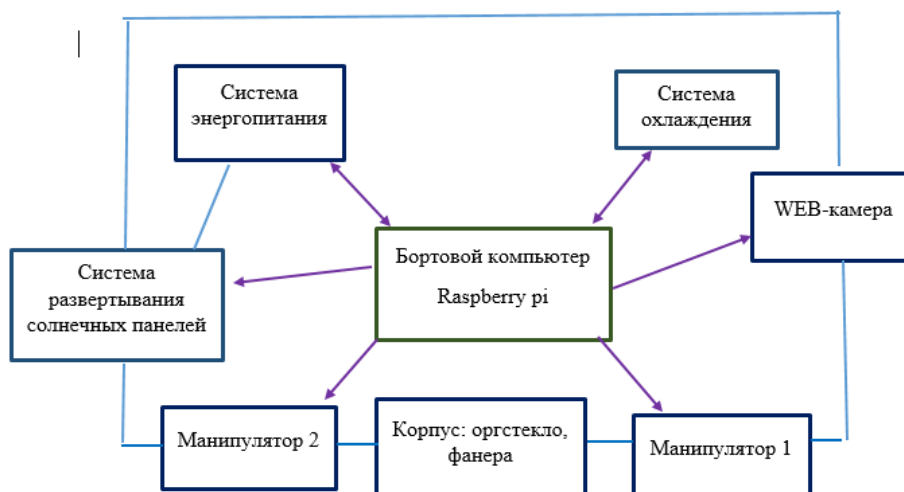


Рис. 1. Структурная схема платформы для внекорабельной деятельности

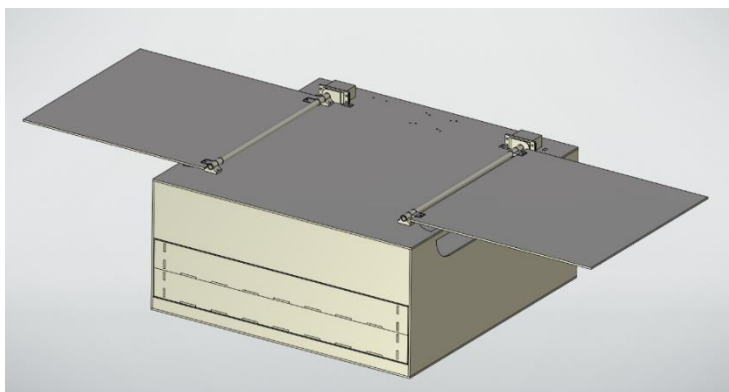


Рис. 2. 3D-модель платформы для внекорабельной деятельности

Основной платой управления платформой является плата raspberry pi. Выбранный способ управления платформой подразумевает использование специализированного сайта, который содержит инструменты для управления платформой. Такой сайт был нами разработан и включает в себя средства считывания и обработки команд и их отправки для обработки на сервер. Данные в клиентской части на сайте конвертируются в специальный формат json и отправляются на сервер на плате raspberry, сервер обрабатывает данные и преобразует их в команды. При выдаче команд ток поступает на определённые gpio контакты, с помощью которых можно управлять моторами постоянного тока.

Понятно, что для обмена данными достаточно просто создать веб-сервер на любом языке, предоставляющих эту возможность. На рисунке 3 приведен пример создания веб-сервера на Node.js.

```
JS server.js > http.createServer() callback
1  var http = require('http')
2
3  http.createServer(function (req, res) {
4    res.writeHead(200, {
5      'Content-Type': 'text/plain',
6      'Access-Control-Allow-Origin': '*'
7    });
8
9    let body = ''
10   req.on('data', chunk => {
11     body += chunk.toString()
12   })
13   req.on('end', () => {
14     console.log(body)
15     res.end('ok')
16   })
17   }).listen(3256, '0.0.0.0')
```

Рис. 3. Пример создания веб-сервера на Node.js

С клиентской части можно подключиться к веб-серверу и отправлять туда данные. На рисунке 4 приведён пример отправки данных на веб-сервер при помощи http на языке javascript.

```
JS client.js > headers > 'Content-Type'
1  fetch('http://localhost:3256', {
2    method: 'POST',
3    headers: {
4      'Content-Type': 'text/plain'
5    },
6    body: JSON.stringify(body)
7  })
```

Рис. 4. Пример отправки данных на веб-сервер через http на javascript

Потом на сервере данные обрабатываются и становятся пригодными для работы, в зависимости от этого можно управлять raspberry pi.